Udacity Machine Learning Engineer Nanodegree
Capstone project proposal: Dog Breed Classifier

Vasileios-Marios Gkortsas

Contents

The Capstone project i will work will be the Dog Breed Classifier.

1 Domain background

The goal of the project is to learn how to build a pipeline to process real-world, user-supplied images. Given an image of a dog, the algorithm will identify an estimate of the canine's breed. If supplied an image of a human, the code will identify the resembling dog breed. The Machine Learning architecture we will use is the Convolution Neural Networks (CNN) one and we will follow 2 training approaches. In the first one, we will create and train the CNN from scratch, In the second one, we will apply transfer learning where we will use a well-known sophisticated architecture together with the corresponding training weights and we will modify and train only the last layer in order to adjust the architecture to our needs.

2 Problem statement

Classification of an image's content is the most common Computer Vision task. We will build different versions of image's classification. The first version will be a crude classifier, where the workflow will detect whether the input image has a human face or a dog. Then, using CNNs we can move to more advanced image classifiers, which will tell us the dog's breed if the image has a dog, or which dog breed the human looks like, if the image has a human face.

We will use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images. But OpenCV does not include a dog classifier, so we will have to use CNNs for dog breed classification. We will create 2 versions of CNNs for dog breed classification, the first one will be a CNN architecture from scratch and the second one will be a pre-trained sophisticated architecture in which we will modify and train only the last layer in order to adjust the predicted number of classes to the number of dog breeds. This approach is called transfer learning.

3 Datasets and inputs

Both the dog and human faces datasets are provided by Udacity and can be either downloaded in the local machine or used directly from the Udacity workspace. We will mainly focus on the dog dataset, since we will train the CNN architecture on this one. The dog dataset folder consists of 3 subfolders named train, valid and test, which will be used in each of the corresponding phases of the workflow. Each of these 3 subfolders has 133 subfolders, where each one of them corresponds to the 133 dog breeds.

The inputs to the CNN will come from these folders after they are converted to Pytorch tensors.

4 Solution statement

The solution that we will build gradually in the provided Jupyter notebook, using PyTorch, consists of the following steps:

- Step 1: Detect Humans
- Step 2: Detect Dogs
- Step 3: Create a CNN to Classify Dog Breeds (from Scratch)
- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)
- Step 5: Write your Algorithm
- Step 6: Test your Algorithm

  Each of these steps has coding assignments noted with a “ TO DO ” label and some questions that need to be answered.

5 Benchmark model

As mentioned above, OpenCV does not provide a dog classifier. Thus, we will use a CNN architecture to detect the various dog breeds. We will train a CNN from scratch, but we will also use pre-trained well known architectures like VGG16 or ResNet50. Since these architectures can classify dog breeds provided in ImageNet, but not all the 133 breeds of our dataset, we will perform transfer learning where we will modify and train the last layer, while all the other layers will be frozen and their weights won’t change.

6 Evaluation metrics

The CrossEntropyLoss function provided by Pytorch will be used to evaluate the train and validation loss of the CNN architecture. The accuracy, defined as the number of times a predicted dog breed label is equal to the reference dog breed label divided by the total number of testing dog images, can be used to estimate the test accuracy of the CNN architecture.

7 Project design

I will implement the project based on the provided description at step 4.